Exploitation des données :

Le Like ne s’applique que sur les chaînes de caractère, il signifie « ressemble à ». Utilise des caractères de substitution :

* % Remplace n’importe quel nombre de caractères, y compris aucun.
* \_ Remplace exactement un caractère.

|  |  |
| --- | --- |
| SELECT… | AVG : efectue la moyenne des valeurs SUM : Effectue la somme des valeurs MIN : retourne la valeur minimale MAX : retourne la valeur maximal COUNT(un\_champs) : retourne le nombre de valeur non nulle COUNT(\*) : retourne le nombre de valeur même nulle COUNT(distinct…) : retourne le nombre devaleur distinct non nulles |
| FROM… | As : change le nom (Produit as P)  …1 INNER JOIN …1 ON …1.id = …2.id |
| WHERE… | < / > / =  ALL : tous les  ANY : un des…  LIKE : ressemble à…  NOT IN : n’est pas…  IN(‘…’,’…’,’…’)  … BETWEEN … AND … |
| GROUP BY… | Regroupe des lignes sur des valeurs identiques (GROUP BY idRayon) |
| HAVING… | Sélectionne des lignes qui ont déjà été agrégées (HAVING count(\*) > 15)  Fonctionne seulement après un GROUP BY |
| ORDER BY… | ASC : croissant  DESC : décroissant |

Curdate() : date d’aujourd’hui  
Years/month/day/day of month : année, mois, jour, jour du mois

|  |  |
| --- | --- |
| INSERT INTO dansQuelleTable( listeDesChamps )  VALUES ( listeDesValeurs ) | INSERT INTO personnel(idPersonnel, nom,prenom,ddn) VALUES (11, 'Dupont', 'Robert', '1958-03-18'); |
| UPDATE nomTable  SET nomChamp=nouvelleValeur WHERE conditionDeMàJ | UPDATE Personnel  SET nom = « DUPONT »  WHERE nom = MARTIN AND prenom = Myriam |
| DELETE FROM nomTable  WHERE conditionSuppression ; | DELETE FROM personnel  WHERE ddn > ‘1980-06-24’  AND nom LIKE “MAR%”; |
| CREATE TABLE nomDeTable(  nomColonne1 typeColonne1\*\*\*,  nomColonne2 typeColonne2\*\*\*,  …  ) | CREATE TABLE caisse(  idCaisse int(10),  DateProd date  …  ); |
| \*\*\* :  **AUTO\_INCREMENT** : Incrémente de 1 la valeur de la colonne à chaque insertion. Souvent utilisé pour la clef primaire.  **NOT** **NULL** : Indique l’obligation de mettre une valeur dans le champ.  **UNIQUE** : Les valeurs dans la colonne doivent toutes être différentes les unes des autres.  **DEFAULT** : Spécifie une valeur par défaut à l’insertion d’une occurrence (ligne) pour le champ en question. | CREATE TABLE caisse(  idCaisse int(10) NOT NULL AUTO\_INCREMENT,  DateProd date NOT NULL DEFAULT ‘2011-09-01’  …  ); |
| CREATE TABLE nomDeTable(  nomColonne1 typeColonne1\*\*\*,  nomColonne2 typeColonne2\*\*\*,  …)  CREATE TABLE nomDeTable(  nomColonne1 typeColonne1\*\*\*,  nomColonne2 typeColonne2\*\*\*,  …) | CREATE TABLE caisse(  idCaisse int(10),  DateProd date  …  Primary key (‘idCaisse”)  );  CREATE TABLE caisse(  IdPassageCaisse int(10) unsigner NOT NULL DEFAULT ‘0’,  idProduit int(10) unsigned NOT NULL DEFAULT ‘0’,  qte int(10) unsigned DEFAULT NULL,  Primary key (‘idPassageCaisse’,’idProduit’)  ); |
|  | check permet de réduire un domaine de définition ici : pour vérifier que la taille d’une personne ne dépasse pas 300 cm |
| CONSTRAINT ‘nom contrainte’ FOREIGN KEY(‘nom’) REFERENCES ‘table a relier (colonne a relier) |  |
| CONSTRAINT ‘nom contrainte’ FOREIGN KEY(‘nom’) REFERENCES ‘table a relier ‘(‘colonne a relier’) ON DELETE… | Cascade : Suppression des occurrences liées set NULL : mise à NULL du champs FK des occurrences liées Restrict : Interdit la supprestion |
| CONSTRAINT ‘nom contrainte’ FOREIGN KEY(‘nom’) REFERENCES ‘table a relier ‘(‘colonne a relier’) ON UPDATE… | CASCADE : MAJ des occurrences liées en cas d’update Restrict : Interdit la maj |
|  | |
| ALTER TABLE caisse  ADD COLUMN pxCaisse FLOAT  ADD COLUMN ajout1 VARCHAR(45) | Ajout, suppression ou modification d’un champ  Ajout, suppression ou modification d’une  contrainte.  Ajoute la colonne pxCaisse de type float  Ajoute la colonne pxCaisse de type varchar |
| ALTER TABLE caisse  MODIFY Column pxCaisse INT  MODIFY Column DateProd DATETIME DEFAULT ‘2011-09- 01’,  DROP COLUMN pxCaisse; | Modifie le champs de pxCaisse en entier Modifie le champs de DateProd en DATETIME par default avec la valeur 2011-09-01  Supprime la colonne pxCaisse |
| ALTER TABLE caisse  ADD CONSTRAINT fk1  FOREIGN KEY (idProduit)  REFERENCES Produit (idProduit)  DROP FOREIGN KEY fk1; | Dans la table caisse  Ajoute la contrainte fk1  en FOREIGN KEY, le champ produit Faisant référence dans la table Produit, du champs idProduit  Supprime la contrainte fk1 |
| DROP TABLE contenir2 | Supprime la table contenir2 |
| RENAME TABLE Rayon TO Rayonnage | Renomme la table Rayon par Rayonnage |
|  | |
| CREATE VIEW v\_ticket\_caisse (numero, totalT)  AS SELECT idPassageCaisse, SUM(qte\*pxUnit)  FROM contenir, PRODUIT  WHERE Produit.idProduit = contenir.idProduit  GROUP BYT idPassageCaisse;  DROP VIEW v\_ticket\_caisse ; | Créé une vue avec les champs numero et totalT  Appel la vue v\_ticket\_caisse ; |
|  |  |
| CREATE USER `userTest` @ `localhost` IDENTIFIED BY `toto`; | Cree un utilisateur du nom de : “userTest”,  qui pourra se connecté seulement a partir : localhost  qui a pour mot de passe : toto |
| DROP USER `userTest` | Supprime l’utilisateur userTest |
| GRANT  SELECT (id, nom, prenom)  ON table absence.etudiant  To userTest@’%’; | DONNE LE DROIT  De selectionner id, le nom et le prenom  de la table absence.etudiant  a userTest. |
| GRANT  ALL Privileges  ON \*.\*/bdToto.\*  TO userTest@’%’ | DONNE LE DROIT  De tout selectionner  dans nimporte quel base/nimporte quel table  A userTest depuis nimporte ou. |
| REVOKE  SELECT (id)  ON table absence.etudiant  To userTest@’%’; | SUPPRIME LE DROIT  De selectionner id  de la table absence.etudiant  a userTest. |
| Flush privileges | Force le SGBD a relire les droits (a mettre apres un grant ou un revoke) |

**TINYINT** : Entier de 0 à 255 (unsigned)

**SMALLINT** : Entier de 0 à 65535 (unsigned)

**MEDIUMINT** : Entier de 0 à 16777215 (unsigned)

**INT** : Entier de 0 à 4294967295 (unsigned)

**BIGINT** : Entier de 0 à 18446744073709551615 (unsigned)

**DECIMAL** : Un nombre à virgule flottante

**DATE** : Une date, va de '1000-01-01' à '9999-12-31'

**DATETIME** : Date et Heure, va de '1000-01-01 00:00:00' à '9999-12-31 23:59:59'

**TIMESTAMP** : Date et Heure exprimée en secondes depuis le 1er janviers 1970. Va de '1970-01-01 00:00:00' à quelque part, durant l'année 2037

**TIME** : Une mesure de l'heure, va de '-838:59:59' à '838:59:59'

**YEAR** : Une année, va de 1901 à 2155

**CHAR** : Chaîne de caractère de taille fixe, va de 1 à 255 caractères

**VARCHAR** : Chaîne de caractère de taille variable, va de 1 à 255 caractères

**TINYTEXT** **ou** **TINYBLOB** : Un objet BLOB ou TEXT, longueur maximale de 255

**TEXT** **ou** **BLOB** : Un objet BLOB ou TEXT, longueur maximale de 65535

**MEDIUMTEXT** **ou** **MEDIUMBLOB** : Un objet BLOB ou TEXT, longueur maximale de 16777215

**LONGTEXT** **ou** **LONGBLOB** : Un objet BLOB ou TEXT, longueur maximale de 4294967295